Automated Retrieval from Multiple Disparate Information Sources; the World Wide Web and the NLM's Sourcerer Project

R. P. Channing Rodgers
Lister Hill National Center for
Biomedical Communications
Building 38A, Room 9S-916
U.S. National Library of Medicine
8600 Rockville Pike
Bethesda MD 20894
Phone: (301)496-9300
Fax: (301)496-0673
E-mail: rodgers@nlm.nih.gov

To appear in a special "Perspectives" issue of the
Journal of the American Society for Information Science,
W. Hersh (Ed.), December 1995.

Abstract

The burgeoning amount of information available via the Internet has heightened awareness of the need for improved tools for resource identification. The U.S. National Library of Medicine's (NLM) Sourcerer project is developing software which accepts a user query, automatically identifies appropriate information resources, and facilitates connection to those sources for information retrieval. The current Sourcerer prototype utilizes the multimedia/multiplatform/multiprotocol network-based hypertext system known as World Wide Web. It also relies upon the knowledge sources of the Unified Medical Language System (UMLS). The UMLS is the result of a long-term project of NLM. It comprises a large Metathesaurus of biomedical concepts (coupled with a semantic network and syntactical/lexical software tools) and the Information Sources Map (ISM), a database of records describing specific biomedical information resources. Recent advances in the standardization of information exchange over computer networks, coupled with the tools provided by UMLS, facilitate query refinement and augmentation, connection to resources, and retrieval from resources. Daunting challenges remain with respect to optimizing resource descriptions, defining optimal algorithms for searching for sources, optimizing user interface design, and organizing retrieved information.

Introduction

The notion of a single readily accessible automated tool to access all of human knowledge is at least as old as H. G. Wells' 1938 book *World Brain*; it resurfaces periodically, as in Vannevar Bush's oft-quoted 1945 article in *Atlantic Monthly*, in which he described a device he called "Memex." Such an application would ideally be vertically integrated, in the sense of accepting natural language queries from its user, and doing whatever is necessary to reformulate the query, locate and connect to an appropriate resource, and return helpfully organized information. It would also be horizontally integrated, in the sense of allowing access to multiple distinct and independent information sources. Although full realization of this tantalizing goal remains beyond the grasp of current technology, recent developments have enabled substantial progress toward it.

Work described here is currently underway at the U.S. National Library of Medicine (NLM). Conducted by multiple groups within NLM, it builds upon and mirrors related efforts being conducted around the globe. Indeed, the synergistic international collaborative efforts enabled by computer network technology, as currently embodied in the Internet, has accelerated both interest in this problem, and progress toward its achievement. The work described is experimental; it remains to be determined how it might fit into the evolution of NLM's online information services. Nothing described here should be taken as an announcement of future services to be offered by NLM.

The NLM is the world's largest library collection devoted to a single professional topic. It has also been a leader in developing online information resources: MEDLINE was one of the earliest computerized bibliographic databases and remains one of the most frequently employed of such services. The number and diversity of NLM online resources has grown steadily over the past decades, and now includes more than 40 databases covering clinical medicine, biomedical research, toxicology and environmental health, health care services, and biotechnology. These databases are delivered by multiple distinct

systems, the two principle ones being a mainframe-based system known as ELHILL , and a PC-based system known as TOXNET .

Central to the work discussed here is a special multi-year initiative of the NLM, known as the Unified Medical Language System™ (UMLS). As initially conceived, the UMLS comprised three knowledge sources (Lindberg, Humphreys, & McCray, 1993):

1) The UMLS Metathesaurus : an attempt to unify the numerous pre-existing defined vocabularies of medicine, the Metathesaurus contains more than 478,562 terms corresponding to over 222,927 distinct concepts, drawn from 31 separate defined vocabularies, including the NLM's own defined vocabulary, Medical Subject Headings, or MeSH .

2) The UMLS Semantic Network (McCray, 1989) includes a list of semantic types (high-level abstract descriptions of meaning) and semantic relations (verbs), as well as a list of meaningful semantic relationships (noun-verb-noun triples composed of two semantic types separated by a semantic relation). Each concept in the Metathesaurus is assigned one or more semantic types.

3) The UMLS Information Sources Map (ISM). As originally conceived, the ISM was to consist of two components: A descriptive portion outlining the nature of various electronically available information resources, and a procedural component providing the means to connect to a resource and retrieve information from it. The descriptive component exists in the form of a database of records describing information sources. Each record describes a single source and contains free-text descriptions, various categorical descriptors that employ elements drawn from lists of pre-defined values (nature of contents, intended audience, etc.), and other descriptors (language, size, update frequency, etc.). Records also include indexing information in the form of semantic types, semantic relationships, and MeSH headings. When indexing articles for the MEDLINE database, the most specific possible index terms are employed. An ISM information source is indexed using broad

terms that capture the essence of its content without spelling it out in full detail.

In 1994, the SPECIALIST™ lexicon was added to the list of UMLS knowledge sources. It contains syntactic information about a subset of Metathesaurus terms as well as English words not appearing in the Metathesaurus. It also includes a set of software tools for lexical manipulation. Over time, syntactical information will be shifted out of the Metathesaurus and into the SPECIALIST lexicon.

The Sourcerer Project and the World Wide Web

Sourcerer is a research application currently under development at NLM. It is intended to accept a user query, automatically identify relevant information sources, and, where possible, connect to the resources and conduct queries on the user's behalf. It builds upon all of the UMLS knowledge sources, most particularly the ISM. Laying out a detailed description of this application will provide a framework for discussing both the issues involved in automated source retrieval and previous and concurrent related work.

Sourcerer adheres to the server-client programming paradigm commonly encountered in network-based software. In this context, a server is a piece of software on one computer which offers services to another piece of software — the client running on the same computer or another computer. The client is often little more than communication software combined with display software. Communication between the server and client programs obeys a set of rules known as a network communications protocol. Sourcerer utilizes the World Wide Web (WWW) (Schatz & Hardin, 1994), a network-based hyper-text system, use of which has grown exponentially since the introduction of the freely available WWW client, NCSA Mosaic, in early 1993. WWW has quickly become one of the most intensively used Internet-based information services. WWW subsumes the capabilities of two contemporaneous network information retrieval tools: It provides a functional superset of the hierarchical menu-driven navigational capabilities of *gopher* (Anklesaria, McCahill, Lindner, Johnson, & Torrey, 1993), and it is easy to graft the

word-based indexing capabilities of Wide-Area Information Servers, or WAIS onto Web applications (Kahle & Medlar, 1991).

WWW presents several features which facilitate the design of a system such as Sourcerer:

1) It is hypertext based. Documents are created using a simple SGML-compliant markup language known as HyperText Markup Language (HTML). HTML allows a component of a document to be marked up as an *anchor*; an anchor is associated with a network address for an informational object such as another HTML file; this address is known as a Uniform Resource Locator (URL). When a HTML documents is displayed, anchors are highlighted; selection of an anchor causes the document specified by the associated URL to be requested.

2) It supports multimedia (where allowed by the underlying hardware platform). Web documents allow the easy intermixing of text and images in a displayed document; downloaded files may contain images, sound, and full-motion animations.

3) It supports multiple communications protocols. The communications protocol underlying WWW is the Hypertext Transport Protocol (HTTP). The Web was designed at the outset to support frequently used communications protocols such as *ftp*, *telnet*, and *gopher*. It was also designed to be readily extensible. The result is a seamless point-and-click graphical front-end which allows the user to forget which communications protocol is actually in use.

4) It is non-proprietary. The HTML and HTTP protocols are defined within the public domain and readily accessible; they are examples of "open standards." They are maintained and developed by organizations such as the Internet Society's Internet Engineering Task Force (IETF) and the newly formed World Wide Web Consortium (W3C), headquartered at MIT in Cambridge and at INSERM in Paris. Any interested party can write software based on these specifications; this had led to an energetic outpouring of Web clients and servers from both public and commercial

sources.

5) It is platform-independent and interoperable. Multiple Web clients are available for the IBM PC, Macintosh, and UNIX environments. The University of Kansas has developed a client for traditional text-based computer terminals. In principle, a client running on any platform should be able to interact successfully with a server running on any platform.

6) It is network-distributed. The user's client can retrieve any document that is being offered by any Web server running on any networks to which the user's computer is attached. This becomes most powerfully evident when the user has access to the globe-spanning network-of-networks known as the Internet (Krol, 1992).

7) It is extensible. Both HTTP and HTML continue to evolve. The National Center for Supercomputer Applications (NCSA) at the University of Illinois has also introduced several important extensibility mechanisms that have been picked up as part of Web standards. The Common Gateway Interface (CGI) allows custom-written applications to use the communications facilities of a Web server. The Common Client Interface (CCI) allows a client-side application to control a local Web client.

8) It provides a high-level scripting language for the creation of graphical user interfaces (GUIs). HTML contains a simple set of commands to specify a form containing text boxes, check boxes, radiobuttons, and selection lists. This point is of particular importance: WWW provides a means of quickly producing a cross-platform user interface. Previously, this required extensive programming, either using software tools native to each target platform, or expensive and complex cross-platform GUI development systems.

As currently implemented, WWW also presents challenging limitations to application software developers:

1) It is stateless. The classical Web transaction involves the delivery of a single document, and proceeds as follows: a client sends a URL to a server, and the server

returns either an error message or a copy of the requested document, and then breaks the connection. Connections are evanescent and although servers often keep statistical logs of file access, there is no active memory of prior transactions. Each request from a client is treated as an independent event. Most traditional database interactions require multiple interactions between user and service: For example, in refining a search expression, segmenting results into tractable subsets, extracting selected results for further processing, etc. This requires that information be carried forward between transactions, a process referred to here as "maintaining state." Some early Web CGI applications attempted to maintain state by coding information into URLs. This was ungainly, and led to confusion when users tried to save and reuse these ephemeral network addresses. To remedy this, a special HTML field known as a "hidden field" was defined as part of the HTML form extensions. A hidden field contains information that is not normally displayed, and is useful for storing state information. Work is in progress on stateful extensions to the HTTP specification.

2) It is insecure. With the currently popular Web clients, it is difficult to be certain of the identity of the parties involved in an information transaction (a process known as authentication), and to prevent the interception of the transmitted material. The latter problem can be addressed by encryption techniques. These problems are being actively addressed by the Web community; several Web clients offer authentication and encryption, though these extensions need to be incorporated into Web standards and more widely propagated.

3) It is evolving rapidly. Some vendors have unilaterally introduced extensions to protocols such as HTML, bypassing the standards process. It is difficult to track the latest developments, to sort out features that are specified in Web standards from those that are unilateral extensions, and to determine when a new feature is widely available in Web clients.

4) HTML forms are limited in capability. The designer has limited control over the appearance and layout of individual forms components, or "widgets" (this is being addressed through the introduction of HTML tables). HTML forms widgets have innately limited functionality: For example, on occasion it would be useful to have the ability to browse a hierarchical set of lists. Many native GUI systems could do so through a system of "walking menus." On a mouse-based system, the user finds an item of interest on a master menu list, selects it, and moves the cursor to the right, whereupon a daughter menu associated with the selected item appears. The process of selection and rightward motion continues until the user finds the item of interest and releases the mouse button. HTML widgets do not support this functionality, and even if they did, the information hierarchy behind such a walking menu system would have to be transmitted with the document, a potentially slow and expensive process. This problem may be addressed in part by Common Client Interface (CCI) applications which allow a client-side application to control a local Web client, and the introduction of downloadable software modules (as currently available in an experimental Web client known as *HotJava;* see Sun Microsystems, 1995).

5) Client implementations are heterogeneous and HTML prevents exact control of the appearance of a document. HTML is a markup language, not a formatting language. The document creator specifies the conceptual parts of a document (titles, paragraphs, anchors, etc.), but client implementors have broad freedom to determine how to represent these objects in a display. Many clients also allow the user to specify font types and sizes, document window size, etc. Furthermore, clients vary as to which version of the evolving HTML standard they implement, and some implementations are incomplete. It is challenging to design a document that will work well on all of the commonly available clients.

6) WWW limits client-side control. At present it is not possible to do things that many

non-Web clients do, such as tailoring server-client interactions based on an editable client-side configuration file, or downloading an entire client-side file as part of a forms-based interaction. This may also be alleviated by new developments such as CCI and HotJava.

7) Uniform Resource Locators are unreliable. When documents are moved or removed, their URLs change. The Web does not allow these changes to be propagated into anchors appearing in documents on remote sites. Deliberations at the IETF have suggested that WWW anchors should employ unique document identifiers known as Uniform Resource Names (URNs) instead of URLs. A URN is the electronic equivalent of the International Standard Book (or Serial) Number. Important details remain to be worked out, such as the designation of a naming authority that would issue URNs and ensure their uniqueness. Document issuers would register valid URLs with an Internet service, and web clients would resort to this service to map the URNs associated with document anchors to a list of one or more URLs, much as the Internet's Domain Name Service (DNS) allows software to resolve a symbolic computer name such as todo.kansas.edu into the unique numerical IP address that networking software actually employs for communication, such as 130.13.33.123 (Albitz & Liu, 1992). This scheme requires that document issuers maintain current records with the URN -> URL resolution service. It is not yet clear which software system will be used for this service; there are numerous competing software schemes for the registration of network-distributed resources, based on both open and proprietary standards.

The Sourcerer Prototype

Sourcerer is written as a CGI application. It is being written in stages, each progressively more sophisticated than its predecessor. The description that follows corresponds to prototype 2 (Rodgers, Srinivasan, & Fullton, 1994). The functional components of Sourcerer are outlined in Figure 1. Sourcerer, functioning behind a WWW server, acts as a server to the end user's WWW client/browser, and as a client in its interactions with four different types of servers. These servers are numbered 1-4 in Figure 1, corresponding to the order in which they are consulted by Sourcerer. Note that numerous arbitrary decisions were made in determining this architecture, and that similar functionality could be achieved by many different schemes.

The sequence of events in using Sourcerer with a mouse-based computer is as follows:

1) The user requests to employ Sourcerer, by specifying the appropriate URL or by clicking on an anchor in a Web document. Sourcerer assigns a unique session identifier to the client, creates a table for state information, and returns the Sourcerer form document (the session identifier is embedded within the form, in a hidden field; hidden fields are used extensively by Sourcerer to minimize the amount of information that the CGI application must maintain on the server side). The user's Web client displays the form.

2) The user enters a search expression by typing a single concept into each of one to three text windows appearing in the form, and submits the form to Sourcerer by selecting a button on the form. As an example, consider the case in which the first window contains "bleeding time" and the second contains "aspirin," as in Figure 2. The two concepts are combined using a Boolean "AND."

3) Sourcerer consults the UMLS knowledge source server (McCray & Razi, 1995). In particular, it determines if any of the concepts entered by the user appear in the Metathesaurus. At present, this is done by a simple exact-pattern match. The next prototype will support lexical variants and partial pattern matching (to allow for

spelling mistakes and differences between British and American usage), and will interact with the user in instances where no matches or multiple matches are found, to help refine the search expression. When a matching concept is found, Sourcerer obtains pertinent information associated with the concept, including synonyms, MeSH headings, and semantic types. In the example search, Sourcerer finds exactly one match for each of the concepts "bleeding time" and "aspirin." If the search expression contains more than one concept, Sourcerer makes a list of all pairwise combinations of the semantic types associated with the concepts and consults the semantic network component of the UMLS knowledge source server to determine what semantic type relations are permissible using these pairings of semantic types. In the example search, aspirin is associated with two semantic types (Organic Chemical and Pharmacological Substance) and bleeding time with one (Diagnostic Procedure). Sourcerer thus composes a list of three potentially meaningful noun-verb-noun triples (exchanging the original search concepts for the semantic types): "Bleeding time assesses effect of aspirin," "bleeding time measures aspirin," and "bleeding time uses aspirin." A form with these triples is returned to the user, who may select any that are appropriate to the search. That portion of the form returned for the example search of Figure 2 appears in Figure 3.

4) Sourcerer now consults the ISM server. Information accrued at the previous stage (including the original search strings, any corresponding Metathesaurus concepts, synonyms, MeSH headings, semantic types, and semantic type relations) are merged into a search expression patterned after the user's original search expression, and used to consult the ISM database. Prototype 2 merges concept groupings with the Boolean "OR," even if the corresponding concepts are grouped with "AND." Prototype 2 searches only the MeSH heading, semantic type, and semantic type relation fields of ISM records. The next prototype will employ all of the ISM fields, including the various categorical and free text fields. Also, it is anticipated that the ISM

server may be folded into the UMLS knowledge source server. The user receives a hypertext document that lists identified information sources, each source appearing as a hypertext anchor.

5) The user may select the anchor associated with any returned source to read a description of the source (drawn from the corresponding ISM database record). The description contains a list of URLs, each of which is an anchor pointing to the online object in question. The next prototype will employ URNs instead, and automatically map the URNs to lists of URLs through a URN -> URL resolution service.

6) The user may connect to a specific information source (referred to as a terminal information source in Fig. 1) by selecting the URL in the hypertext description page corresponding to that source. In the case of a WWW-based experimental MEDLINE access system (see below), the original user search expression is automatically passed along, so that the query form for that service is already filled in for the user. The result is a form similar to the one in Figure 2. Simply clicking on a retrieval button triggers a MEDLINE search and returns bibliographic citations. The first of a set of 20 out of 41 citations so retrieved appears in Figure 4.

A companion application, *Apprentice*, allows new ISM records to be created using an interactive HTML form. When completed, this will allow information providers to register their services with the ISM system. Human indexers will augment these records with appropriate MeSH headings, semantic types, and semantic type relations.

Related NLM Developments, and Future Prototypes

The experimental WWW-based MEDLINE application just described is under development at the NLM by Lawrence Kingsland and associates. It builds upon earlier experience with a PC-based system, COACH, that employs the UMLS Metathesaurus to interactively assist the user in refining the search expression (Kingsland, Harbourt, Syed,

& Schuyler, 1993). This system is tailored to MEDLINE and relies upon MeSH, the defined vocabulary used for indexing MEDLINE, but it points the way to more general aids for guided searching, based on the full set of vocabularies within the Metathesaurus. The interface between Sourcerer and this application, which (like the state mechanism of Sourcerer) relies on HTML hidden fields, demonstrates a simple form of interprocess communication for Web applications.

The long term goal of the Sourcerer project is to provide a software aid that helps a user interactively refine a search expression, and then, with minimal further action on the part of the user, identifies appropriate resources, connects to them, retrieves pertinent information, and returns the information to the user in a usefully organized manner. The current prototype requires three stages of interaction:

1) The user provides an initial search expression.

2) The user selects appropriate semantic type relations from a list proposed by Sourcerer.

3) Once the ISM database has been consulted, the user selects and then initiates connections to appropriate sources. These sources may require further interaction.

Although the present state of the art precludes Sourcerer from completely eliminating further interactions with the operator after the first two (search refinement) steps, it is clearly desirable to minimize them. One impediment to automated connection to information sources is that the procedure for querying an information source is often unique to that source. Another evolving open standard addresses this issue. Z39.50 (ANSI/NISO, 1994), originally designed for interaction with bibliographic databases but later generalized to other types of information, provides a standard software interface to databases that allows a user query to be sent to multiple information servers, without the need to write interface software that is unique to each service. The next Sourcerer prototype will make use of a recently developed experimental Z39.50 interface to MEDLINE.

Pertinent Work by Others: The Yale ISM Application, and WWW Resource Identification Schemes

The first application of the ISM was created by Dr. Perry Miller and his collaborators at Yale University (Miller, Frawley, Wright, Roderer, & Powsner, 1995). Written in LISP and running on a UNIX workstation, the server component of the system is accessed as one of multiple services available via a campus information system known as NetMenu. The client-side software relies on a proprietary communications package (Dynacomm) which to date is available only under the Microsoft Windows operating system. The client presents the user with a forms-based front end which includes text windows for the user's query, and various buttons that allow the user to specify the type of information sources sought, their mode(s) of availability, and their intended use(s). The application returns a list of potentially useful sources, along with controls that allow the user to read more about specific sources, regroup the returned information in various ways, and to connect to specific sources. The Yale group has also collaborated in an attempt to transplant the technology to another campus. Their service receives 200-300 connections per month, a small fraction of the 15-20,000 connections received by the NetMenu system.

The problem of resource identification looms large in the World Wide Web community. The browsing paradigm enabled by hypertext can be quite useful for a small number of documents; it is nearly useless for answering a precisely defined question under tight time constraints. Throughout the work to be discussed two fundamental questions arise: how should the retrieval systems be organized? (centrally organized vs. distributed and loosely coupled); and, how should indexing be done? (human-generated vs. automated).

The first attempt to facilitate focused retrievals over the Web appeared in the form of manually created and maintained subject-based lists of sources, such as the NCSA Meta-Index (National Center for Supercomputer Applications, 1995) Planet Earth (Naval Command, Control & Ocean Surveillance Center, 1995), and Yahoo (Yahoo, 1995).

Another approach to resource identification has been the automated creation of indices

based on robotic searches of the Web. Software processes repetitively follow hypertext links, accumulating information along the way. The predecessor of such systems on the Internet is *Archie*, a system which catalogs files accessible via anonymous ftp (Emtage & Deutsch, 1992). A similar system, *Veronica*, exists for *gopher* resources. These systems vary according to the level of detail of indexing. The WWW Worm (McBryan, 1994) creates a searchable index using the titles and URLs of all the documents it visits. The Repository-Based Software Engineering (RBSE) JumpStation system indexes titles and top-level headings, while Web Crawler (Pinkerton, 1994) and the RBSE Spider (Eichmann, 1994) index the full text of HTML documents; the latter is coupled with a user interface that supports relevance-feedback searching (Salton & Buckley, 1990).

Cataloging robots are appealing in their simplicity and their ability to work within the Web as it currently exists, but have raised concerns about both ethics (tying up remote resources, potentially impairing their use by others) and their technical limitations (employing network resources to download documents, and then discarding most of what is retrieved; centralizing information at a single point of failure). The Harvest project (Bowman, Danzig, Hardy, Manber, & Schwartz, 1994) addresses these and other concerns by creating a distributed indexing scheme, which, however, requires a level of administrative effort and between-site coordination not yet seen within the Web community. Content descriptions are stored in a format known as the Summary Object Interchange Format (SOIF). Another experiment (Dodge, Marx, & Pfeiffenberger, 1995) has created catalogs based on the documents stored in local caching (or "proxy") servers (local computers which temporarily archive the Web documents that are retrieved by local users, in case they are requested by another local user). Yet another approach relies on the creation of a network-distributed hierarchy of content descriptions which allows the user to refine a query, navigating toward an appropriate informational end-node (Sheldon, Duda, Weiss, & Gifford, 1995).

The more speculative relative of the simple Web robot is the software agent. For

example, the Internet Softbot (Etzioni & Weld, 1994) uses information available on the Web. It presents the user with a graphical user interface that helps specify the goal of the user (for example, to send mail to a person with a specified family name at a specified institution), and then employs tools from the discipline known as machine learning to determine how to achieve this goal. The discussion of software robots predates invention of WWW by a number of years; their lack of practical impact to date may be due in part to their potential complexity, and in part to concerns about security.

The human-generated catalogs of the traditional paper-based library have also provided a model for Web-based information indexing. GENVL (McBryan, 1994) is a registry system which allows information providers to create small catalog entries (containing information such as title, purpose, and author) describing the materials they have to offer. ALIWEB (Koster, 1994) uses simple catalog templates defined by the Internet Anonymous FTP Archives (IAFA) working group of the IETF. This meta-information is indexed on the machine that actually contains the information, and multiple local indices are periodically harvested by a central host and merged into a common searchable database. Harvest SOIF entries also would be commonly human-generated.

Increasing attention is being paid to formalizing the description of resource content (Neuss & Kent, 1995). This information-about-information has come to be called metadata. The IETF URI working group has added discussion of Uniform Resource Citations (URCs) to that of URNs and URLs. OCLC has sponsored a metadata workshop and proposed various methods for encoding metadata (Vizine-Goetz, Godby, & Bendig, 1995). Dobson & Burrill (1995) have proposed extensions to HTML that allow description of content, to facilitate extraction of this information into a database. Bellcore (Shklar, Shah, & Basu, 1995) has developed an object-oriented retrieval system based on a scheme for the description of metadata (InfoHarness Repository Definition Language, or IRDL).

Discussion

An automated source identification and retrieval system built on the ISM application faces subtle communications, user interface, and information retrieval problems, as pointed out by the Yale experience. Relying as it does on Internet-based standards that appeared after the initiation of the Yale project, the Sourcerer prototype avoids certain of the communications problems faced at Yale, including the problems of cross-platform support (inherent in WWW), connection to sources (inherent in use of the Internet), and retrieval from sources (if a limited number of retrieval protocols, such as Z39.50, are used by the sources). The need to write and maintain a communications script for each source is an obstacle to scaling the Yale system to a large number of sources. Although reliance on open standards has reduced the communications problems faced by the Sourcerer prototype, it carries a price in the burden of tracking and participating in the standardization process of a rapidly evolving discipline.

Sourcerer still faces the same problems as the Yale system with respect to interface design and information retrieval (consult Miller et al., 1995, for an excellent discussion). As development begins on the next prototype, some of the issues under consideration are:

1) Designing a user interface that accommodates the needs of diverse users. Some users will want to minimize intermediate interactions with the ISM application; others will benefit from a series of repetitive interactions in which the retrieval set is optimized. Some users will want to select sources to be queried, while others will want to jump as quickly as possible from their original query to results from terminal information sources. Some information sources will accommodate automated queries, others will almost certainly require operator interaction. Although it will be straightforward to take into account preferences such as source language, it will be more challenging to balance the user's desire for minimal interactions against the user's budget, where certain sources are freely available and others charge access fees. Organizing and displaying retrieved information poses problems. Ideally,

duplicate items should be eliminated. If the user requests automated retrieval from terminal information sources, it may be desirable to conduct searches of the sources in parallel, allowing results to return at different times. The mode of presentation will likely depend both on the background skills of the user, as well as the nature and number of resources identified by the search. Strategies under consideration include ranking of retrievals; categorization of retrievals according to information type, availability, or location; combinations of ranking and categorization; and clustering of related sources in two- and three-dimensional graphical displays.

2) Optimizing the design and content of ISM resource descriptions, and the algorithms for retrieval of sources from the ISM. The present content of ISM records was defined in an *ad hoc* fashion; precision and recall will be determined by both the ISM indexing methods and the algorithms used to fashion a query for the ISM, based on the original user query. The goals associated with retrieval of a record from the ISM will generally differ from those associated with retrieval from a terminal information source. As pointed out by the Yale workers, if a user is searching for information about "A AND B" (where "AND" is the Boolean operation), it may make more sense to query the ISM using "A OR B," sending along "A AND B" to the terminal information sources. If post-filtering methods can be applied to results returned from terminal information sources, it is likely that, with respect to the ISM query, it is preferable to pay a penalty in terms of reduced precision in order to maximize recall. Masys (1992) performed a study of the precision and recall associated with various ways of combining the MeSH, semantic type, and semantic type relation index terms of ISM records. This study demonstrated high recall (99%) and low precision (47%) when using "OR" to combine these index terms.

   An equally subtle issue arises in trying to determine how to formulate a search of the ISM by combining the myriad information that is returned from consulting the UMLS using elements of the user's original query, and how to rank the results that

are returned. Given a sufficiently large set of actual user queries, with a corresponding list of optimal sources from the ISM, it would be possible in principle to apply a neural network to help optimize the algorithms discussed here. Assembling the required query test set would be a substantial undertaking.

References

Albitz, P., & Liu, C. (1992). *DNS and BIND*. Sebastopol CA: O'Reilly & Associates, Inc.

Anklesaria, F., McCahill, M., Lindner, P., Johnson, D., & Torrey, D. (1993, March) *F.Y.I. on the Internet Gopher Protocol (a distributed document search and retrieval protocol).* Memorandum, University of Minnesota (refer to gopher://boombox.micro.umn.edu/00/gopher/gopher_protocol/DRAFT_Gopher_FYI_RFC.txt).

ANSI/NISO (1994). *ANSI Z39.50: Information Retrieval Service and Protocol* (refer to the URL ftp://ftp.loc.gov/pub/z3950/).

Bowman, C.M., Danzig, P., Hardy, D.R., Manber, U., & Schwartz, M.F. (1994). The Harvest information discovery and access system. In I. Goldstein, J. Hardin, T. Berners-Lee, L. Brandt, R. Cailliau, J. Fullton, T. Krauskopf, E. Krol, Y. Kikuta, B. Kucera, C. Moore, R. Rodgers, M. Schwartz, Y. Rubinsky, T. Rutkowski, J. Stewart, M. Tenenbaum, & J. Thot-Thompson (Eds.), *Proceedings of the Second International World-Wide Web Conference* (pp. 763-772). Chicago: NCSA.

Bush, V. (1945). As we may think. *Atlantic Monthly*, *176*, 101-108.

Dobson, S.A., & Burrill, V.A. (1995, April). *Lightweight databases*. Paper presented at the Third International World-Wide Web Conference, Darmstadt.

Dodge, C., Marx, B., & Pfeiffenberger, H. (1995, April). *Web cataloguing through cache exploitation and steps toward consistency maintenance*. Paper presented at the Third International World-Wide Web Conference, Darmstadt.

Eichmann, D. (1994). The RBSE spider — Balancing effective search against Web load. In R. Cailliau, O. Nierstrasz & M. Ruggier (Eds.), *Proceedings of the First International World-Wide Web Conference* (pp. 113-120). Geneva: CERN.

Emtage, A., & Deutsch, P. (1992). *Archie — an electronic directory service for the Internet.* USENIX Association Winter Conference Proceedings (pp. 93-110). San Francisco: USENIX.

Etzioni, O., & Weld, D. (1994). A Softbot-based interface to the Internet. *Communications of the ACM, 37,* 72-79.

Kahle, B., & Medlar, A. (1991, April). *An information system for corporate users: wide area information servers*. (Technical report TMC-199.) Cambridge MA: Thinking Machines, Inc.

Kingsland, L.C., Harbourt, A.M., Syed, E.J., & Schuyler, P.L. (1993). Coach™: Applying UMLS Knowledge Sources in an expert searcher environment. *Bulletin of the Medical Library Association, 81,* 178-183.

Koster, M. (1994). ALIWEB — Archie-like indexing in the Web. In R. Cailliau, O. Nierstrasz, & M. Ruggier (Eds.), *Proceedings of the First International World-Wide Web Conference* (pp. 91-100). Geneva: CERN.

Krol, E. (1992). *The Whole Internet User's Guide & Catalog.* Sebastopol CA: O'Reilly & Associates, Inc.

Lindberg D.A.B., Humphreys, B.L., & McCray A.T. (1993). The Unified Medical Language System. *Methods of Information in Medicine, 32,* 281-291.

Masys, D.R. (1992). An evaluation of the source selection elements of the prototype UMLS information sources map. In Mark E. Frisse (Ed.), *Proceedings of the Sixteenth Annual Symposium on Computer Applications in Medical Care* (pp. 295-298). Baltimore, MD: American Medical Informatics Association.

McBryan, O.A. (1994). GENVL and WWWW: Tools for taming the Web. In R. Cailliau, O. Nierstrasz, & M. Ruggier (Eds.), *Proceedings of the First International World-Wide Web Conference* (pp. 79-90). Geneva: CERN.

McCray A.T. (1989). The UMLS semantic network. In L.C. Kingsland (Ed.),

Proceedings of the 13th Annual Symposium on Computer Applications in Medical Care. (p. 503-507). Washington, DC: IEEE Computer Society Press.

McCray, A.T. & Razi A. (1995) The UMLS knowledge source server. In R.A. Greenes, H.E. Peterson, & D.J. Protti (Eds.), *Proceedings of the Eighth World Congress on Medical Informatics: Medinfo '95* (pp. 144-147). Vancouver: International Medical Informatics Association.

Miller P.L., Frawley S.J., Wright L., Roderer N.K., & Powsner S.M. (1995). Lessons learned from a pilot implementation of the UMLS information sources map. *Journal of the American Medical Informatics Association, 2*, 102-115.

National Center for Supercomputer Applications (1995). *NCSA MetaIndex* (refer to URL http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/MetaIndex.html).

Naval Command, Control & Ocean Surveillance Center (1995). *Planet Earth* (refer to URL http://white.nosc.mil/info.html).

Neuss, C., & Kent, R.E. (1995, April). *Conceptual analysis of resource meta-information*. Paper presented at the Third International World-Wide Web Conference, Darmstadt.

Pinkerton, B. (1994). Finding what people want: Experiences with the WebCrawler. In I. Goldstein, J. Hardin, T. Berners-Lee, L. Brandt, R. Cailliau, J. Fullton, T. Krauskopf, E. Krol, Y. Kikuta, B. Kucera, C. Moore, R. Rodgers, M. Schwartz, Y. Rubinsky, T. Rutkowski, J. Stewart, M. Tenenbaum, & J. Thot-Thompson (Eds.), *Proceedings of the Second International World-Wide Web Conference* (pp. 821-829). Chicago: NCSA.

Rodgers R.P.C., Srinivasan S., & Fullton J. (1994, October). *Sourcerer: Thesaurus-assisted automated source identification for the World-Wide Web.* Paper presented at the Second International World-Wide Web Conference, Chicago (refer to the URL http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Searching.html).

Salton, G., & Buckley, C. (1990). Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science, 41*, 288-297.

Schatz, B.R., & Hardin, J.B. (1994). NCSA Mosaic and the World Wide Web: Global hypermedia protocols for the Internet. *Science, 265*, 895-901.

Sheldon, M.A., Duda, A., Weiss, R., & Gifford, D.K. (1995, April). *Discover: A resource discovery system based on content routing*. Paper presented at the Third International World-Wide Web Conference, Darmstadt.

Shklar, L., Shah, K., & Basu, C. (1995, April). *Putting legacy data on the Web: A repository definition language*. Paper presented at the Third International World-Wide Web Conference, Darmstadt.

Sun Microsystems, Inc. (1995). *The HotJava browser: A white paper*. (refer to the URL http://java.sun.com/1.0alpha2/doc/overview/hotjava/index.html).

Vizine-Goetz, D., Godby, J., & Bendig, M. (1995, April). *Spectrum: A Web-based tool for describing electronic resources*. Paper presented at the Third International World-Wide Web Conference, Darmstadt.

Wells, H.G. (1938). *World Brain*. Garden City, NY: Doubleday Doran.

Yahoo (1995). *Yahoo* (refer to URL http://www.yahoo.com/).

Figure 1. Functional Diagram of Sourcerer.

Sourcerer is a Common Gateway Interface (CGI) application operating behind a World Wide Web server. It saves information on disk to support stateful interactions with the user via any fully forms-capable WWW client such as NCSA Mosaic. It communicates with four other types of servers: the UMLS knowledge source server, the ISM server, a location server (which maps URNs to URLs), and multiple terminal information sources.

Figure 2. Sourcerer Prototype (Version 2) Query Page.

The search expression entered here reads "bleeding time AND aspirin" (where AND is the Boolean operator).

Figure 3. Sourcerer Prototype (Version 2) Semantic Type Relation Selection List.

The prototype consulted the Metathesaurus to determine the semantic types corresponding to concepts matching the strings "bleeding time" and "aspirin;" it then consulted the UMLS semantic network to determine what semantic type relations (non-verb-noun triples) were considered to be potentially meaningful. The user is then asked to select any of these expressions that appears relevant to the original query. In this instance, the user has selected the first and third relations.

Figure 4. Sourcerer Prototype (Version 2) Citation Retrieved from an Experimental Web-Based Version of MEDLINE as Accessed via Sourcerer.

One of 41 citations retrieved, this article is the first in a batch of 20 that were down-loaded. Note that the term "aspirin" appears in the title, and the term "bleeding time" appears highlighted in the abstract.

Brief Biographical Sketch

R. P. C. Rodgers is responsible for the Information Sources Map Project, one of several components of the NLM's Unified Medical Language System (UMLS). A clinical pathologist with a strong interest in the application of computers to biomedical needs, he was an early adapter of World Wide Web technology. He developed HyperDOC, the NLM's WWW server, and (in collaboration with S. Srinivasan) OnLine Images (OLI), a system for Web-based retrieval of large catalogued image archives. He currently chairs the NSF/NCSA World Wide Web Federal Consortium, is a member of the International World Wide Web Conference Committee (IW3C2), and serves on several NLM steering groups related to the development of new information systems. In his capacity as a member of the IW3C2, he has been responsible for organizing the live transmission of audio and video from the International WWW Conferences, using the experimental Internet-based MBONE multicasting system.